Dear Bob,

Thankyou for your letter, and your Bulletin. I think I get the idea of the role you want to play, and I will try to tailor my future correspondence to fit.

The dissassembler program from the bulletin is quite good, it works well, and provides binary output split into "split-octal" groups. Given the time-honored 8080 traditions, this should be quite valuable. Enclosed is a program I wrote to carry out the same task, in hexadecimal. It is longer, but much faster.

&(10) may be used to monitor the operation of the Basic by typing &(10)=205. This causes the computer to display alternate bits of the memory after address19903 (where the Basic usually stops). Thus you can see the contents the variables A-Z, of the other addresses used by Basic, and of the the addresses used by the operating system after 20265.

The question of how to enter machine-language routines is an interesting one. The Z-80 does not have the "natural" split-octal codes of the 8080, although its set of instructions ~~xxxxxx~~ includes the 8080"s. The manufacturers recommend hex, but this is virtually impossible to assemble by hand. Decimal sounds like the worst possible choice, but it is what the Basic uses, and it is what I use.

To employ a machine language sub- routine, first it must be written in something like Assembly language. Then the Assembly program must be translated (by hand) into some number sytem. I generally translate first into binary because my Manual (Mostek) makes this most convenient. The Binary can then be converted to decimal (a hand calculator helps here).

At this point, the program consists of a sequence of addresses, each paired with a number between 000 and 255. Now, the program may be entered lowest address to highest, with a sequence of commands

%(address)=NNN

where 'address' is in decimal, and NNN is the progam data (also in decimal).

The Basic converts NNN into binary and stores it at the specified address. Unfortueately, it also changes address+1. In this case, it would set address+1 to zero. Thus, the lowest-to-highest order for program entry.

Or, the entry process can be simplified by use of a program, an example of which is enclosed.

Please note that you can not enter a machine-language routine at the start of the editor-buffer, because this space is written into by any key depressed while a program is not executing. Thus, by using 20200, you can enter approximately 20 characters without disturbing your program. This is enough to enter the program statements (not more than 15 or so characters), or the give commands (RUN, List, CALL), or edit short statements. The Basic never clears the editor buffer except on resets, it merely writes and re-writes, using the first encountered carriage-return to signal the end of the durrent command.

Finally, I have managed to get my memory-expansion device working. Since this was the first home-electronics project that I ever undertook, I decided to keep things extremely simple, but to protect the processor to the utmost. Thus, my expansinn consists of address and data buffers, address decoding, and 2114 static memories. 2102's could be used for about half the cost-per-bit, but would require four times the board space and (most importantly) four times the number of connections.

The actual interface is very simple; The Bally PA-1 Service manual lists the pin-out of the main edge-connector, and all of the Z-80 address, data, and control signals come out, along with the Arcade"s own signals. The addresses are de-coded in a straightforward way; my device recognizes an 8-K block, which is selected by a DIP-swithh. The address-select is ANDed with the MemReq, and a true result turns on the data buffers (which are normally tri-stated), and the 3-to-8 line decoder, which employs the next three address lines (A12-A10) to select 1K of memory. I will eventually use seven of these blocks for memory (this project is on a tight budget), and the eigth is for various special purposes.

I did discover one very importand fact. The Multi-plexer for the Custom IC's appears todecode the address bus, and presumably tri-states for memory addresses outside its range. However,this does not appear to be so. My unit simply refused to function properly until the BUSOFF line was brought low by the above-mentioned conjunction pf the proper address and memreq.

To date, I have installed 2K of extra memory, and find it a great advantage, mostly because its contents are preserved when the Basic is reset. However, I am still ata very early stage, and haven't gotten any software which takes advantage of the new memory yet.

Sincerely,

John Sweeney

John H. Sweeney

MICRO-BOX 0
MEMORY EXPANSION